

## IN THE CLAIMS

Please amend claim 1, 5, 6, 8, 9, 12, 13, 15, 18, and 19 as indicated below:

1. (Currently Amended) A method comprising:  
  
identifying a source code block in source code;  
  
transforming the source code block including transforming references to values stored in memory locations into references to corresponding values stored in registers associated with the memory locations; and  
  
generating compensation native code to update each memory location with a value from an associated register to provide native code, corresponding to a source code exception handler associated with the source code block, access to the updated memory locations.
2. (Original) The method of claim 1, comprising generating exception-handling code to handle any exception not handled by the native code corresponding to existing source code exception handlers.
3. (Original) The method of claim 1, wherein the compensation native code forms part of the native code corresponding to an exception handler.
4. (Original) The method of claim 1 further comprising generating native code to initialize the registers; and generating register-initialization exception-handling native code to handle exceptions occurring during register initialization.
5. (Currently Amended) The method of claim 4, wherein the register-initialization exception-handling native code allows native code corresponding to the ~~untransformed~~ source code block to be executed if an error occurs during register initialization.
6. (Currently Amended) The method of claim 1, wherein the source code is in the form of byte codes compiled by a JAVA compiler.
7. (Original) The method of claim 1, performed by a Just-In-Time compiler.

8. (Currently Amended) A machine-readable medium that provides instructions, which when executed by a processor, cause the processor to perform operations comprising:

identifying a source code block in source code;

transforming the source code block, including transforming references to values stored in memory locations into references to corresponding values stored in registers associated with the memory locations; and

generating compensation native code to update each memory location with a value from an associated register to provide native code, corresponding to a source code exception handler associated with the source code block, access to the updated memory locations.

9. (Currently Amended) The machine-readable medium of claim 8, wherein said ~~plurality of~~ instructions when executed further cause said processor to perform operations comprising generating exception-handling code to handle any exception not handled by the native code corresponding to the existing source with exception handlers.

10. (Original) The machine-readable medium of claim 8, wherein the compensation native code forms part of the native code corresponding to an exception handler.

11. (Original) The machine-readable medium of claim 8, further comprising generating native code to initialize the registers; and generating register-initialization exception-handling native code to handle exceptions occurring during register initialization.

12. (Currently Amended) The machine-readable medium of claim 11, wherein the register-initialization exception-handling native code allows native code corresponding to the ~~untransformed~~ source code block to be executed if an error occurs during register initialization.

13. (Currently Amended) The machine-readable medium of claim 8, wherein the source code is in the form of byte codes compiled by a JAVA compiler.
14. (Original) The machine-readable medium of claim 8, wherein the operations are performed by a Just-In-Time compiler.
15. (Currently Amended) An apparatus for compiling source code into native code comprising:
- ~~a processor and a memory coupled thereto; and~~
- a memory coupled to the processor, the memory comprising instructions corresponding to an identifier to identify a source code block in source code; a transformer to transform the source code block, including transforming references to values stored in memory locations into references to corresponding values stored in registers associated with the memory locations; and a compensator to generate compensation native code to update each memory location with a value from an associated register to provide native code, corresponding to a source code exception handler associated with the source code block, access to the updated memory locations.
16. (Original) The apparatus of claim 15 comprising a generator to generate exception-handling code to handle any exception not handled by the native code corresponding to existing source code exception handlers.
17. (Original) The apparatus of claim 15 wherein the compensation native code forms part of the native code corresponding to an exception handler.
18. (Currently Amended) The apparatus of claim 15 in which the ~~optimizer~~ transformer generates native code to initialize the registers and register-initialization exception-handling native code to handle exceptions occurring during register initialization.
19. (Currently Amended) The apparatus of claim 18 wherein the register-initialization exception-handling native allows native code corresponding to the

untransformed source code to be executed if an error occurs during register initialization.